FortiOS™ Handbook - IPv6

VERSION 5.2.7

FORTINET DOCUMENT LIBRARY

http://docs.fortinet.com

FORTINET VIDEO GUIDE

http://video.fortinet.com

FORTINET BLOG

https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT

https://support.fortinet.com

http://cookbook.fortinet.com/how-to-work-with-fortinet-support/

FORTIGATE COOKBOOK

http://cookbook.fortinet.com

FORTINET TRAINING SERVICES

http://www.fortinet.com/training

FORTIGUARD CENTER

http://www.fortiguard.com

END USER LICENSE AGREEMENT

http://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK

Email: techdocs@fortinet.com

Wednesday, June 22, 2016

FortiOS™ Handbook - IPv6

01-520-112805-20141016

TABLE OF CONTENTS

Change Log

Date	Change Description
2016-06-22	Correction to CLI syntax for IPv6 addresses
2014-10-16	CLI syntax update.
2014-06-13	FortiOS 5.2 major release.
2013-12-01	Official release.

Introduction

The origins of Internet Protocol Version 6 (IPv6) date back to December 1998 with the publication of RFC 2460, which describes IPv6 as the successor to IPv4, the standard communications protocol still in use by the majority of users today. This transition away from IPv4 was a direct response to the foreseeable exhaustion of 32-bit IPv4 addresses, which are virtually all but assigned—all 4.3 billion.

IPv4 uses 32-bit addresses, which means that there is a theoretical address limit of 2 to the power of 32. The IPv6 address scheme is based on a 128-bit address, resulting in a theoretical address limit of 2 to the power of 128.

Possible addresses:

IPv4 = Roughly 4.3 billion

IPv6 = Over 340 undecillion (340 followed by 36 digits)

Assuming a world population of approximately 8 billion people, IPv6 would allow for each individual to have approximately 42,535,295,865,117,200,000,
000,000,000 devices with an IP address. That's 42 quintillion devices, so it's unlikely that we will ever need to worry about the availability of IPv6 addresses.

Aside from the difference of possible addresses, there is also the different formatting of the addresses. A computer would view an IPv4 address as a 32-bit string of binary digits made up of 1s and 0s, broken up into 4 octets of 8 digits separated by a period:

```
10101100.00010000.11111110.00000001
```

To make the number more user-friendly, we translate the address into decimal, again 4 octets separated by a period:

```
172.16.254.1
```

A computer would view an IPv6 address as a 128-bit string of binary digits made up of 1s and 0s, broken up into 8 octets of 16 digits separated by a colon:

```
0010000000000001:0000110110111:0000000000000000:000000000000010:0000000000000000:000000000
    0000000:0000000000000000:0000000000100000
```

To make this number a little more user-friendly, we translate it into hexadecimal, again 8 octets separated by a colon, for example:

```
2001:0db8:0000:0002:0000:0000:0000:0020
```

We can further simplify the above address. Because any four-digit group of zeros within an IPv6 address may be reduced to a single zero or altogether omitted, the above address can be reduced to:

```
2001:0db8:0000:0002:0:0:0:20
```

or

```
2001:db8:0:2::20
```

IPv6 packet structure

Each IPv6 packet consists of a mandatory fixed header and optional extension headers, and carries a payload, which is typically either a datagram and/or Transport Layer information. The payload could also contain data for the Internet Layer or Link Layer. Unlike IPv4, IPv6 packets aren't fragmented by routers, requiring hosts to implement Maximum Transmission Unit (MTU) Path Discovery for MTUs larger than the smallest MTU (which is 1280 octets).

Jumbograms and jumbo payloads

In IPv6, packets which exceed the MTU of the underlying network are labelled jumbograms, which consist of a jumbo payload. A jumbogram typically exceeds the IP MTU size limit of 65,535 octets, and provides the jumbo payload option, which can allow up to nearly 4GiB of payload data, as defined in RFC 2675. When the MTU is determined to be too large, the receiving host sends a 'Packet too Big' ICMPv6 type 2 message to the sender.

Fragmentation and reassembly

As noted, packets that are too large for the MTU require hosts to perform MTU Path Discovery to determine the maximum size of packets to send. Packets that are too large require a 'Fragment' extension header, to divide the payload into segments that are 8 octets in length (except for the last fragment, which is smaller). Packets are reassembled according to the extension header and the fragment offset.

Benefits of IPv6

Some of the benefits of IPv6 include:

- More efficient routing
- Reduced management requirement
- Stateless auto-reconfiguration of hosts
- Improved methods to change Internet Service Providers
- Better mobility support
- Multi-homing
- Security
- Scoped address: link-local, site-local, and global address space

IPv6 Features

In order to configure IPv6 features using the web-based manager, IPv6 must be enabled using Feature Select. Go to **System > Config > Features**, enable IPv6, and click **Apply**.

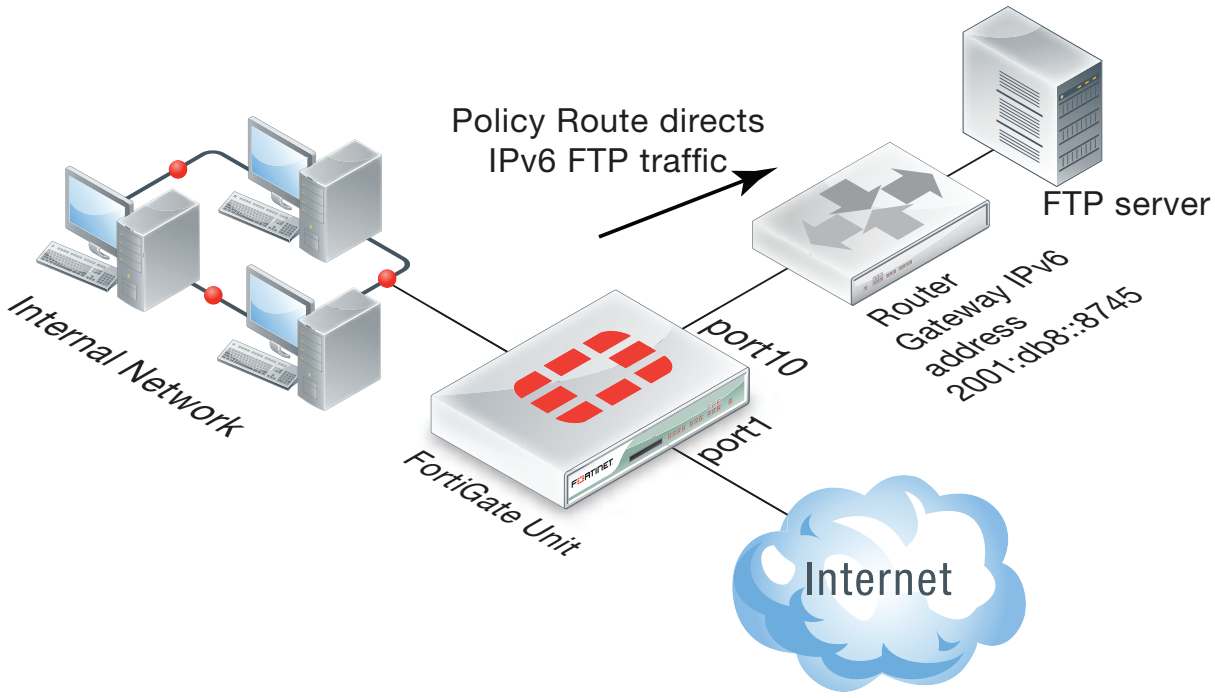
The following IPv6 features are available from the FortiOS web manager:

IPv6 policies

IPv6 security policies are created both for an IPv6 network and a transitional network. A transitional network is a network that is transitioning over to IPv6 but must still have access to the Internet or must connect over an IPv4 network.

These policies allow for this specific type of traffic to travel between the IPv6 and IPv4 networks. The IPv6 options for creating these policies is hidden by default. You must enable this feature under **System > Config > Features**.

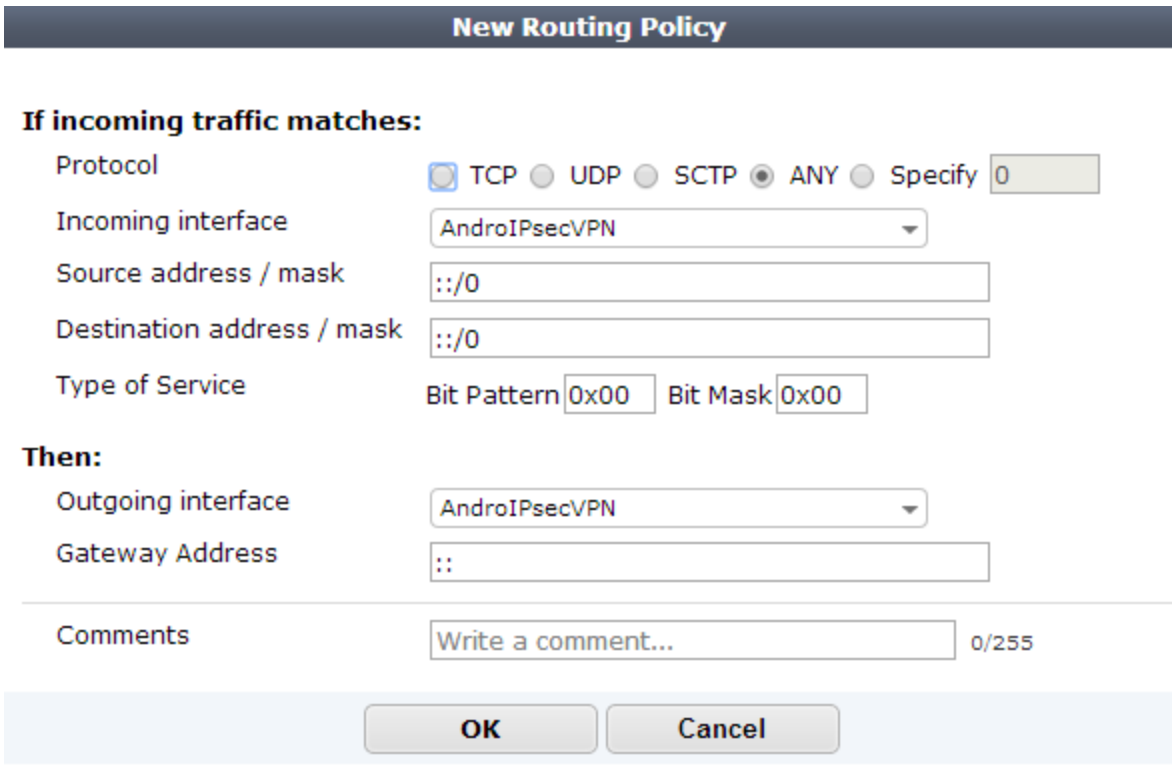
IPv6 policy route



IPv6 policy routing

IPv6 policy routing functions in the same was as IPv4 policy routing. To add an IPv6 policy route, go to **Router > Static > Policy Routes** and select **Create New > IPv6 Policy Route**.

Adding an IPv6 Policy route



You can also use the following command to add IPv6 policy routes:

```
config router policy6
   edit 0
      set input-device <interface>
      set src <ipv6_ip>
      set dst <ipv6_ip>
      set protocol <0-255>
      set gateway <ipv6_ip>
      set output-device <interface>
      set tos <bit_pattern>
      set tos-mask <bit_mask>
   end
```

IPv6 security policies

IPv6 security policies support all the features supported by IPv4 security policies:

- Policy types and subtypes.
- NAT support including using the destination interface IP address, fixed port, and dynamic IP pools.
- All security features (antivirus, web filtering, application control, IPS, email filtering, DLP, VoIP, and ICAP).
- All traffic shaping options, including: shared traffic shaping, reverse shared traffic shaping, and per-IP traffic shaping.
- All user and device authentication options.

IPv6 Policy Monitor

Once policies have been configured and enabled it is useful to be able to monitor them. To get an overview about what sort of traffic the policies are processing, go to **Policy & Objects > Monitor > IPv6 Policy Monitor**.

IPv6 explicit web proxy

You can use the explicit web proxy for IPv6 traffic. To do this you need to:

- Enable the Explicit Proxy from the dashboard.
- Enable the IPv6 explicit web proxy from the CLI.
- Enable the explicit web proxy for one or more FortiGate interfaces. These interfaces also need IPv6 addresses.
- Add IPv6 web proxy security policies to allow the explicit web proxy to accept IPv6 traffic.

Use the following steps to set up a FortiGate unit to accept IPv6 traffic for the explicit web proxy at the Internal interface and forward IPv6 explicit proxy traffic out the wan1 interface to the Internet.

1. Go to **System > Dashboard > Status** and turn on **Explicit Proxy** under the **Features > Security Features** widget. Click **Apply**.
2. Enter the following CLI command to enable the IPv6 explicit web proxy:

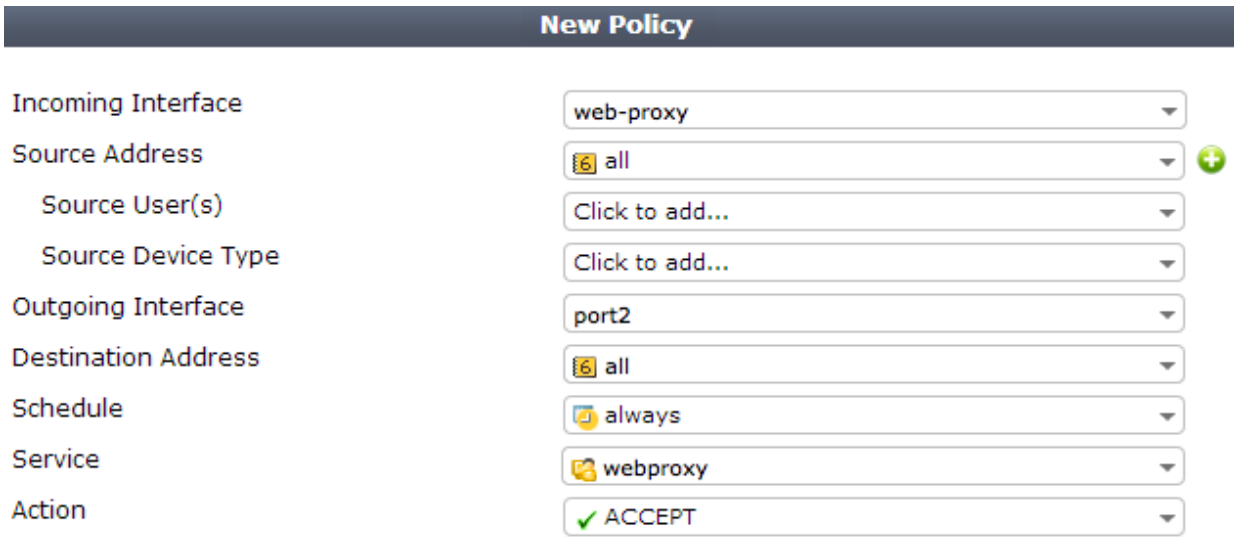
```
config web-proxy explicit
   set status enable
   set ipv6-status enable
end
```

3. Go to **System > Network > Interfaces** and edit the **internal** interface, select **Enable Explicit Web Proxy** and select **OK**.
4. Go to **Policy & Objects > Policy > IPv6** and select **Create New** to add an IPv6 explicit web proxy security policy with the following settings shown.

   This IPv6 explicit web proxy policy allows traffic from all IPv6 IP addresses to connect through the explicit web proxy and through the wan1 interface to any IPv6 addresses that are accessible from the wan1 interface.

If you have enabled both the IPv4 and the IPv6 explicit web proxy, you can combine IPv4 and IPv6 addresses in a single explicit web proxy policy to allow both IPv4 and IPv6 traffic through the proxy.

Example IPv6 Explicit Web Proxy security policy



Restricting the IP address of the explicit IPv6 web proxy

You can use the following command to restrict access to the IPv6 explicit web proxy using only one IPv6 address. The IPv6 address that you specify must be the IPv6 address of an interface that the explicit HTTP proxy is enabled on. You might want to use this option if the explicit web proxy is enabled on an interface with multiple IPv6 addresses.

For example, to require users to connect to the IPv6 address 2001:db8:0:2::30 to connect to the explicit IPv6 HTTP proxy, use the following command:

```
config web-proxy explicit
   set incoming-ipv6 2001:db8:0:2::30
end
```

Restricting the outgoing source IP address of the IPv6 explicit web proxy

You can use the following command to restrict the source address of outgoing web proxy packets to a single IPv6 address. The IP address that you specify must be the IPv6 address of an interface that the explicit HTTP proxy is enabled on. You might want to use this option if the explicit HTTP proxy is enabled on an interface with multiple IPv6 addresses.

For example, to restrict the outgoing packet source address to 2001:db8:0:2::50:

```
config http-proxy explicit
   set outgoing-ip6 2001:db8:0:2::50
end
```

VIP64

VIP64 policies can be used to configure static NAT virtual IPv6 address for IPv4 addresses. VIP64 can be configured from the CLI using the following commands:

```
config firewall vip64
   edit <zname_str>
      set arp-reply {enable | disable}
```

```
        set color <color_int>
        set comment <comment_str>
        set extip <address_ipv6>[-address_ipv6]
        set extport <port_int>
        set id <id_num_str>
        set mappedip [<start_ipv4>-<end_ipv4>]
        set mappedport <port_int>
        set portforward {enable | disable}
        set src-filter <addr_str>
    end
```

VIP64 CLI Variables and Defaults

Variable	Description	Default
`<zname_str>`	Enter the name of this virtual IP address.	No default.
`arp-reply {enable | disable}`	Select to respond to ARP requests for this virtual IP address.	enable
`color <color_int>`	Enter the number of the color to use for the group icon in the web-based manager.	0
`comment <comment_str>`	Enter comments relevant to the configured virtual IP.	No default.
`extip <address_ipv6>[-address_ipv6]`	Enter the IP address or address range on the external interface that you want to map to an address or address range on the destination network. If `mappedip` is an IP address range, the FortiGate unit uses `extip` as the first IP address in the external IP address range, and calculates the last IP address required to create an equal number of external and mapped IP addresses for one-to-one mapping. To configure a dynamic virtual IP that accepts connections destined for any IP address, set `extip` to ::.	::

Variable	Description	Default
extport <port_int>	Enter the external port number that you want to map to a port number on the destination network. This option only appears if portforward is enabled. If portforward is enabled and you want to configure a static NAT virtual IP that maps a range of external port numbers to a range of destination port numbers, set extport to the first port number in the range. Then set mappedport to the start and end of the destination port range. The FortiGate unit automatically calculates the end of the extport port number range.	0
id <id_num_str>	Enter a unique identification number for the configured virtual IP. Not checked for uniqueness. Range 0 - 65535.	No default.
mappedip [<start_ipv4>-<end_ipv4>]	Enter the IP address or IP address range on the destination network to which the external IP address is mapped. If mappedip is an IP address range, the FortiGate unit uses extip as the first IP address in the external IP address range, and calculates the last IP address required to create an equal number of external and mapped IP addresses for one-to-one mapping. If mappedip is an IP address range, the FortiGate unit uses extip as a single IP address to create a one-to-many mapping.	0.0.0.0

Variable	Description	Default
`mappedport <port_int>`	Enter the port number on the destination network to which the external port number is mapped. You can also enter a port number range to forward packets to multiple ports on the destination network. For a static NAT virtual IP, if you add a map to port range the FortiGate unit calculates the external port number range.	0
`portforward {enable | disable}`	Select to enable port forwarding. You must also specify the port forwarding mappings by configuring `extport` and `mappedport`.	disable
`src-filter <addr_str>`	Enter a source address filter. Each address must be in the form of an IPv4 subnet (x:x:x:x:x:x:x:x/n). Separate addresses with spaces.	null

VIP46 policies can be used to configure static NAT virtual IPv4 address for IPv6 addresses. VIP46 can be configured from the CLI using the following commands (see the table below for variable details):

```
config firewall vip46
   edit <name_str>
      set arp-reply {enable | disable}
      set color <color_int>
      set comment <comment_str>
      set extip <address_ipv4>[-address_ipv4]
      set extport <port_int>
      set id <id_num_str>
      set mappedip [<start_ipv6>-<end_ipv6>]
      set mappedport <port_int>
      set portforward {enable | disable}
      set src-filter <add_str>
   end
```

VIP46 CLI Variables and Defaults

Variable	Description	Default
`<name_str>`	Enter the name of this virtual IP address.	No default.
`arp-reply {enable | disable}`	Select to respond to ARP requests for this virtual IP address.	enable

Variable	Description	Default
`color <color_int>`	Enter the number of the color to use for the group icon in the web-based manager.	0
`comment <comment_str>`	Enter comments relevant to the configured virtual IP.	No default.
`extip <address_ipv4>[-address_ipv4]`	Enter the IP address or address range on the external interface that you want to map to an address or address range on the destination network. If `mappedip` is an IP address range, the FortiGate unit uses `extip` as the first IP address in the external IP address range, and calculates the last IP address required to create an equal number of external and mapped IP addresses for one-to-one mapping. To configure a dynamic virtual IP that accepts connections destined for any IP address, set `extip` to 0.0.0.0.	0.0.0.0
`extport <port_int>`	Enter the external port number that you want to map to a port number on the destination network. This option only appears if `portforward` is enabled. If `portforward` is enabled and you want to configure a static NAT virtual IP that maps a range of external port numbers to a range of destination port numbers, set `extport` to the first port number in the range. Then set `mappedport` to the start and end of the destination port range. The FortiGate unit automatically calculates the end of the `extport` port number range.	0
`id <id_num_str>`	Enter a unique identification number for the configured virtual IP. Not checked for uniqueness. Range 0 - 65535.	No default.

Variable	Description	Default
`mappedip [<start_ipv6>-<end_ipv6>]`	Enter the IP address or IP address range on the destination network to which the external IP address is mapped. If mappedip is an IP address range, the FortiGate unit uses `extip` as the first IP address in the external IP address range, and calculates the last IP address required to create an equal number of external and mapped IP addresses for one-to-one mapping. If `mappedip` is an IP address range, the FortiGate unit uses `extip` as a single IP address to create a one-to-many mapping.	::
`mappedport <port_int>`	Enter the port number on the destination network to which the external port number is mapped. You can also enter a port number range to forward packets to multiple ports on the destination network. For a static NAT virtual IP, if you add a map to port range the FortiGate unit calculates the external port number range.	0
`portforward {enable | disable}`	Select to enable port forwarding. You must also specify the port forwarding mappings by configuring `extport` and `mappedport`.	disable
`src-filter <addr_str>`	Enter a source address filter. Each address must be in the form of an IPv4 subnet (x.x.x.x/n). Separate addresses with spaces.	null

IPv6 Network Address Translation

NAT66, NAT64, and DNS64 are now supported for IPv6. These options provide IPv6 NAT and DNS capabilities withIPv6-IPv4 tunnelling or dual stack configurations. The commands are available only in the CLI.

Fortinet supports all features described in RFC 6146. However, for DNS64 there is no support for handling Domain Name System Security Extensions (DNSSEC). DNSSEC is for securing types of information that are

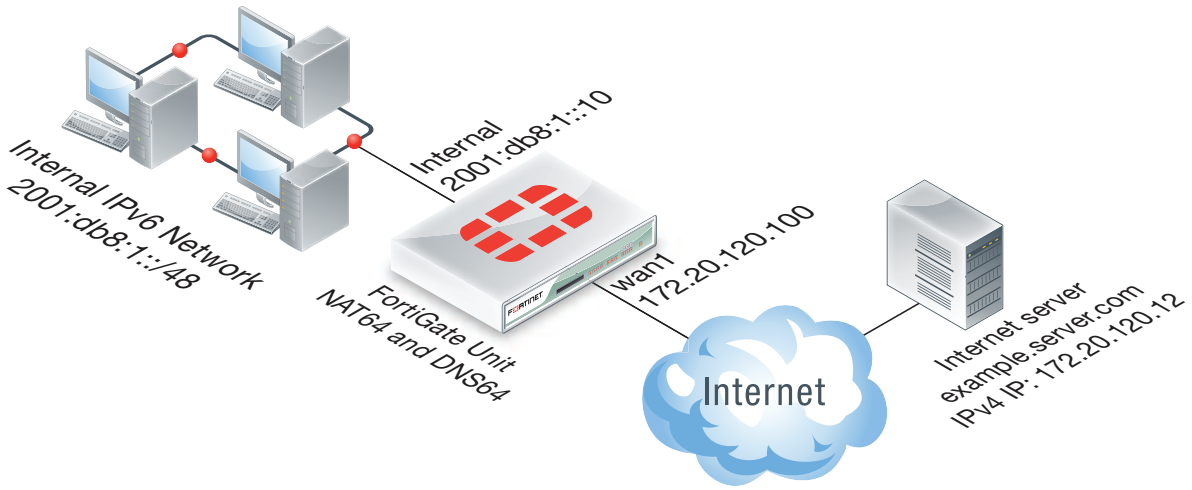
provided by the DNS as used on an IP network or networks. You can find more information about DNS64 in RFC 6147.

NAT64 and DNS64 (DNS proxy)

NAT64 is used to translate IPv6 addresses to IPv4 addresses so that a client on an IPv6 network can communicate transparently with a server on an IPv4 network.

NAT64 is usually implemented in combination with the DNS proxy called DNS64. DNS64 synthesizes AAAA records from A records and is used to synthesize IPv6 addresses for hosts that only have IPv4 addresses. 'DNS proxy' and 'DNS64' are interchangeable terms.

Example NAT64 configuration



With a NAT64 and DNS64 configuration in place on a FortiGate unit, clients on an IPv6 network can transparently connect to addresses on an IPv4 network. NAT64 and DNS64 perform the IPv4 to IPv6 transition, allowing clients that have already switched to IPv6 addresses to continue communicating with servers that still use IPv4 addresses.

To enable NAT64 and DNS64, use the following CLI commands:

Enable NAT64

```
config system nat64
   set status enable
end
```

Enable the DNS proxy on the IPv6 interface

```
config system dns-server
   edit internal
   end
```

In your DHCP6 configuration, configure the IPv6 interface IP address as the DNS6 server IP address. The FortiGate will proxy DNS requests to the system DNS server.

```
config system dhcp6 server
   edit 1
```

```
      set interface internal
      config ip-range
         edit 1
            set start-ip 2001:db8:1::11
            set end-ip 2001:db8:1::20
         end
      set dns-server1 2001:db8:1::10
   end
```

NAT64 policies

You can configure security policies for NAT64 using the web-based manager. For these options to appear, the feature must be enabled using **Feature Select**. You can then configure the policies under **Policy & Objects > Policy > NAT64**.

NAT64 policies and can also be configured from the CLI using the following command:

```
config firewall policy64
```

In the following section, you will configure a NAT64 policy that allows connections from an internal IPv6 network to an external IPv4 network.

Configuring NAT64 to allow a host on the IPv6 network to connect to the Internet server

In this example, the Internal IPv6 network address is 2001:db8:1::/48 and the external IPv4 network address is 172.20.120.0/24. NAT64 is configured to allow a user on the internal network to connect to the server at IPv4 address 172.20.120.12. In this configuration, sessions exiting the wan1 interface must have their source address changed to an IPv4 address in the range 172.20.120.200 to 172.20.120.210.

Enter the following command to enable NAT64:

```
config system nat64
   set status enable
end
```

Enabling NAT64 with the `config system nat64` command means that all IPv6 traffic received by the current VDOM can be subject to NAT64 if the source and destination address matches an NAT64 security policy.

By default, the setting `always-synthesize-aaaa-record` is enabled. If you disable this setting, the DNS proxy (DNS64) will attempt to find an AAAA records for queries to domain names and therefore resolve the host names to IPv6 addresses. If the DNS proxy cannot find an AAAA record, it synthesizes one by adding the NAT64 prefix to the A record.

By using the `nat64-prefix` option of the `config system nat64` command to change the default nat64 prefix from the well-known prefix of 64:ff9b::/96 and setting `always-synthesize-aaaa-record` to `enable` (default), the DNS proxy does not check for AAAA records but rather synthesizes AAAA records.

As an alternative to the above entry, there is the optional configuration that would allow the resolution of CNAME queries.

```
config system nat64
   set status enable
   set nat64-prefix 64:ff9b::/96
   set always-synthesize-aaaa-record enable
end
```

Enter the following command to add an IPv6 firewall address for the internal network:

```
config firewall address6
```

```
    edit internal-net6
       set ip6 2001:db8:1::/48
    end
```

Enter the following command to add an IPv4 firewall address for the external network:

```
config firewall address
   edit external-net4
      set subnet 172.20.120.0/24
      set associated-interface wan1
   end
```

Enter the following command to add an IP pool containing the IPv4 address that the should become the source address of the packets exiting the wan1 interface:

```
config firewall ippool
   edit exit-pool4
      set startip 172.20.120.200
      set endip 172.20.120.210
   end
```

Enter the following command to add a NAT64 policy that allows connections from the internal IPv6 network to the external IPv4 network:

```
    config firewall policy64
       edit 0
          set srcintf internal
          set srcaddr internal-net6
          set dstintf wan1
          set dstaddr external-net4
          set action accept
          set schedule always
          set service ANY
          set logtraffic enable
          set ippool enable
          set poolname exit-pool4
       end
```

The `srcaddr` can be any IPv6 firewall address and the `dstaddr` can be any IPv4 firewall address.

Other NAT64 policy options include `fixedport`, which can be used to prevent NAT64 from changing the destination port. You can also configure traffic shaping for NAT64 policies.

How a host on the internal IPv6 network communicates with example.server.com that only has IPv4 address on the Internet

1. The host on the internal network does a DNS lookup for example.server.com by sending a DNS query for an AAAA record for example.server.com.
2. The DNS query is intercepted by the FortiGate DNS proxy.
3. The DNS proxy attempts to resolve the query with a DNS server on the Internet and discovers that there are no AAAA records for example.server.com.
4. The previous step is skipped if `always-synthesize-aaaa-record` is enabled.
5. The DNS proxy performs an A-record query for example.server.com and gets back an RRSet containing a single A record with the IPv4 address 172.20.120.12.

6. The DNS proxy then synthesizes an AAAA record. The IPv6 address in the AAAA record begins with the configured NAT64 prefix in the upper 96 bits and the received IPv4 address in the lower 32 bits. By default, the resulting IPv6 address is 64:ff9b::172.20.120.12.
7. The host on the internal network receives the synthetic AAAA record and sends a packet to the destination address 64:ff9b::172.20.120.12.
8. The packet is routed to the FortiGate internal interface where it is accepted by the NAT64 security policy.
9. The FortiGate unit translates the destination address of the packets from IPv6 address 64:ff9b::172.20.120.12 to IPv4 address 172.20.120.12 and translates the source address of the packets to 172.20.120.200 (or another address in the IP pool range) and forwards the packets out the wan1 interface to the Internet.

NAT66

NAT66 is used for translating an IPv6 source or destination address to a different IPv6 source or destination address. NAT66 is not as common or as important as IPv4 NAT, as many IPv6 addresses do not need NAT66 as much as IPv4 NAT. However, NAT66 can be useful for a number of reasons. For example, you may have changed the IP addresses of some devices on your network but want traffic to still appear to be coming from their old addresses. You can use NAT66 to translate the source addresses of packets from the devices to their old source addresses.

In FortiOS, NAT66 options can be added to an IPv6 security policy from the CLI. Configuring NAT66 is very similar to configuring NAT in an IPv4 security policy. For example, use the following command to add an IPv6 security policy that translates the source address of IPv6 packets to the address of the destination interface (similar to IPv4 source NAT:

```
config firewall policy6
   edit 0
      set srcintf internal
      set dstintf wan1
      set srcaddr internal_net
      set dstaddr all
      set action accept
      set schedule always
      set service ANY
      set nat enable
   end
```

Its also can be useful to translate one IPv6 source address to another address that is not the same as the address of the exiting interface. You can do this using IP pools. For example, enter the following command to add an IPv6 IP pool containing one IPv6 IP address:

```
config firewall ippool6
   edit example_6_pool
      set startip 2001:db8::
      set endip 2001:db8::
   end
```

Enter the following command to add an IPv6 firewall address that contains a single IPv6 IP address.

```
config firewall address6
   edit device_address
      set ip6 2001:db8::132/128
   end
```

Enter the following command to add an IPv6 security policy that accepts packets from a device with IP address 2001:db8::132 and translates the source address to 2001:db8::.

```
config firewall policy6
   edit 0
      set srcintf internal
      set dstintf wan1
      set srcaddr device_address
      set dstaddr all
      set action accept
      set schedule always
      set service ANY
      set nat enable
      set ippool enable
      set poolname example_6_pool
   end
```

NAT66 destination address translation

NAT66 can also be used to translate destination addresses. This is done in an IPv6 policy by using IPv6 virtual IPs. For example, enter the following command to add an IPv6 virtual IP that maps the destination address 2001:db8::dd to 2001:db8::ee.

```
config firewall vip6
   edit example-vip6
      set extip 2001:db8::dd
      set mappedip 2001:db8::ee
   end
```

Enter the following command to add an IPv6 security policy that accepts packets with a destination address 2001:db8::dd and translates that destination address to 2001:db8::ee.

```
config firewall policy6
   edit 0
      set srcintf internal
      set dstintf wan1
      set srcaddr all
      set dstaddr example-vip6
      set action accept
      set schedule always
      set service ANY
   end
```

NAT64 and NAT66 session failover

The FortiGate Clustering Protocol (FGCP) supports IPv6, NAT64, and NAT66 session failover. If session pickup is enabled, these sessions are synchronized between cluster members and, after an HA failover, the sessions will resume with only minimal interruption.

NAT46

NAT46 is used to translate IPv4 addresses to IPv6 addresses so that a client on an IPv4 network can communicate transparently with a server on an IPv6 network.

To enable NAT46, use the following CLI command:

```
config firewall vip46
```

NAT46 policies

Security policies for NAT46 can be configured from the web-based manager. For these options to appear in the web-based manager, this feature must be enabled using **Feature Select**. You can then configure the policies under **Policy & Objects > Policy > NAT46**.

NAT46 policies and can also be configured from the CLI using the following command:

```
config firewall policy46
```

ICMPv6

Internet Control Message Protocol version 6 (ICMPv6) is the new implementation of the Internet Control Message Protocol (ICMP) that is part of Internet Protocol version 6 (IPv6). The ICMPv6 protocol is defined in RFC 4443.

ICMPv6 is a multipurpose protocol. It performs such things as:

- error reporting in packet processing
- diagnostic functions
- Neighbor Discovery process
- IPv6 multicast membership reporting

It is also designed as a framework to use extensions for use with future implementations and changes.

Examples of extensions that have already been written for ICMPv6:

- Neighbor Discovery Protocol (NDP) - a node discovery protocol in IPv6 which replaces and enhances functions of ARP.
- Secure Neighbor Discovery Protocol (SEND) - an extension of NDP with extra security.
- Multicast Router Discovery (MRD) - allows discovery of multicast routers.
- ICMPv6 messages use IPv6 packets for transportation and can include IPv6 extension headers. ICMPv6 includes some of the functionality that in IPv4 was distributed among protocols such as ICMPv4, ARP (Address Resolution Protocol), and IGMP (Internet Group Membership Protocol version 3).
- ICMPv6 has simplified the communication process by eliminating obsolete messages.

ICMPv6 messages are subdivided into two classes: error messages and information messages.

Error Messages are divided into four categories:

- Destination Unreachable
- Time Exceeded
- Packet Too Big
- Parameter Problems
- Information messages are divided into three groups:
- Diagnostic messages
- Neighbor Discovery messages
- Messages for the management of multicast groups.

ICMPv6 Types and Codes

ICMPv6 has a number of messages that are identified by the “Type” field. Some of these types have assigned “Code” fields as well. The table below shows the different types of ICMP Types with their associated codes if there are any.

Type codes 0 − 127 are error messages and type codes 128 − 255 are for information messages.

ICMPv6 Types and Codes

Type #	Type Name	Code
0	Reserved	0 - no route to destination 1 - communication with destination administratively prohibited 2 - beyond scope of source address 3 - address unreachable 4 - port unreachable 5 - source address failed ingress/egress policy 6 - reject route to destination 7 - Error in Source Routing Header
1	Destination Unreachable	
2	Packet Too Big	
3	Time Exceeded	0 - hop limit exceeded in transit 1 - fragment reassembly time exceeded
4	Parameter Problem	0 - erroneous header field encountered 1 - unrecognized Next Header type encountered 2 - unrecognized IPv6 option encountered
100	Private Experimentation	
101	Private Experimentation	
102 - 126	Unassigned	

Type #	Type Name	Code
127	Reserved for expansion if ICMPv6 error messages	
128	Echo Request	
129	Echo Replay	
130	Multicast Listener Query	
131	Multicast Listener Report	
132	Multicast Listener Done	
133	Router Solicitation	
134	Router Advertisement	
135	Neighbor Solicitation	
136	Neighbor Advertisement	
137	Redirect Message	
138	Router Renumbering	0 - Router Renumbering Command 1 - Router Renumbering Result 255 - Sequence Number Reset
139	ICMP Node Information Query	0 - The Data field contains an IPv6 address which is the Subject of this Query. 1 - The Data field contains a name which is the Subject of this Query, or is empty, as in the case of a NOOP. 2 - The Data field contains an IPv4 address which is the Subject of this Query.
140	ICMP Node Information Response	0 - A successful reply. The Reply Data field may or may not be empty. 1 - The Responder refuses to supply the answer. The Reply Data field will be empty. 2 - The Qtype of the Query is unknown to the Responder. The Reply Data field will be empty.

Type #	Type Name	Code
141	Inverse Neighbor Discovery Solicitation Message	
142	Inverse Neighbor Discovery Advertisement Message	
143	Version 2 Multicast Listener Report	
144	Home Agent Address Discovery Request Message	
145	Home Agent Address Discovery Reply Message	
146	Mobile Prefix Solicitation	
147	Mobile Prefix Advertisement	
148	Certification Path Solicitation Message	
149	Certification Path Advertisement Message	
150	ICMP messages utilized by experimental mobility protocols such as Seamoby	
151	Multicast Router Advertisement	
152	Multicast Router Solicitation	
153	Multicast Router Termination	
154	FMIPv6 Messages	
155	RPL Control Message	
156	ILNPv6 Locator Update Message	
157	Duplicate Address Request	
158	Duplicate Address Confirmation	
159 – 199	Unassigned	
200	Private experimentation	

Type #	Type Name	Code
201	Private experimentation	
255	Reserved for expansion of ICMPv6 informational messages	

IPv6 in dynamic routing

Unless otherwise stated, routing protocols apply to IPv4 addressing. This is the standard address format used. However, IPv6 is becoming more popular and new versions of the dynamic routing protocols have been introduced.

As with most advanced routing features on your FortiGate unit, IPv6 settings for dynamic routing protocols must be enabled before they will be visible in the GUI. To enable IPv6 configuration in the GUI, enable it in **System > Config > Features**. Alternatively, you can directly configure IPv6 for RIP, BGP, or OSPF protocols using CLI commands.

Dual stack routing

Dual stack routing implements dual IP layers in hosts and routers, supporting both IPv6 and IPv4. A dual stack architecture supports both IPv4 and IPv6 traffic and routes the appropriate traffic as required to any device on the network. Administrators can update network components and applications to IPv6 on their own schedule, and even maintain some IPv4 support indefinitely if that is necessary. Devices that are on this type of network, and connect to the Internet, can query Internet DNS servers for both IPv4 and IPv6 addresses. If the Internet site supports IPv6, the device can easily connect using the IPv6 address. If the Internet site does not support IPv6, then the device can connect using the IPv4 addresses.

In FortiOS, dual stack architecture it is not comprised merely of basic addressing functions that operate in both versions of IP. The other features of the appliance, such as UTM and routing, can also use both IP stacks.

If an organization with a mixed network uses an Internet service provider that does not support IPv6, they can use an IPv6 tunnel broker to connect to IPv6 addresses that are on the Internet. FortiOS supports IPv6 tunnelling over IPv4 networks to tunnel brokers. The tunnel broker extracts the IPv6 packets from the tunnel and routes them to their destinations.

IPv6 tunnelling

IPv6 Tunnelling is the act of tunnelling IPv6 packets from an IPv6 network through an IPv4 network to another IPv6 network. Unlike NAT, once the packet reaches its final destination, the true originating address of the sender will still be readable. The IPv6 packets are encapsulated within packets with IPv4 headers, which carry their IPv6 payload through the IPv4 network.

The key to IPv6 tunnelling is the ability of the two devices to be dual stack compatible in order to work with both IPv4 and IPv6 at the same time. In the process, the entry node of the tunnel portion of the path will create an

encapsulating IPv4 header and transmit the encapsulated packet. The exit node at the end of the tunnel receives the encapsulated packet, removes the IPv4 header, updates the IPv6 header, and processes the packet.

There are two types of tunnels in IPv6:

Automatic tunnels: Automatic tunnels are configured by using IPv4 address information embedded in an IPv6 address – the IPv6 address of the destination host includes information about which IPv4 address the packet should be tunnelled to.

Configured tunnels: Configured tunnels must be configured manually. These tunnels are used when using IPv6 addresses that do not have any embedded IPv4 information. The IPv6 and IPv4 addresses of the endpoints of the tunnel must be specified.

Tunnel configuration

There are a few ways in which the tunnelling can be performed depending on which segment of the path between the endpoints of the session the encapsulation takes place.

Host to Host: Dual Stack capable hosts that are interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. In this case, the tunnel spans the entire path taken by the IPv6 packets.

Network Device to Host: Dual Stack capable network devices can tunnel IPv6 packets to their final destination IPv6 or IPv4 host. This tunnel spans only the last segment of the path taken by the IPv6 packets.

The node that does the encapsulation needs to maintain soft state information about each tunnel in order to process the IPv6 packets.

Use the following command to tunnel IPv6 traffic over an IPv4 network. The IPv6 interface is configured under `config system interface`. The command to do the reverse is `config system ipv6-tunnel`. These commands are not available in Transparent mode.

```
config system sit-tunnel
   edit <tunnel name>
      set destination <tunnel _address>
      set interface <name>
      set ip6 <address_ipv6>
      set source <address_ipv4>
   end
```

Variable	Description	Default
`edit <tunnel_name>`	Enter a name for the IPv6 tunnel.	No default.
`destination <tunnel_ address>`	The destination IPv4 address for this tunnel.	0.0.0.0
`interface <name>`	The interface used to send and receive traffic for this tunnel.	No default.
`ip6 <address_ipv6>`	The IPv6 address for this tunnel.	No default.
`source <address_ipv4>`	The source IPv4 address for this tunnel.	0.0.0.0

Tunnelling IPv6 through IPsec VPN

A variation on tunnelling IPv6 through IPv4 is to use an IPsec VPN tunnel between two FortiGate devices. FortiOS supports IPv6 over IPsec. In this sort of scenario, two networks using IPv6 behind FortiGate units are separated by the Internet, which uses IPv4. An IPsec VPN tunnel is created between the FortiGate units and a tunnel is created over the IPv4-based Internet, but the traffic in the tunnel is IPv6. This has the additional advantage of securing the traffic.

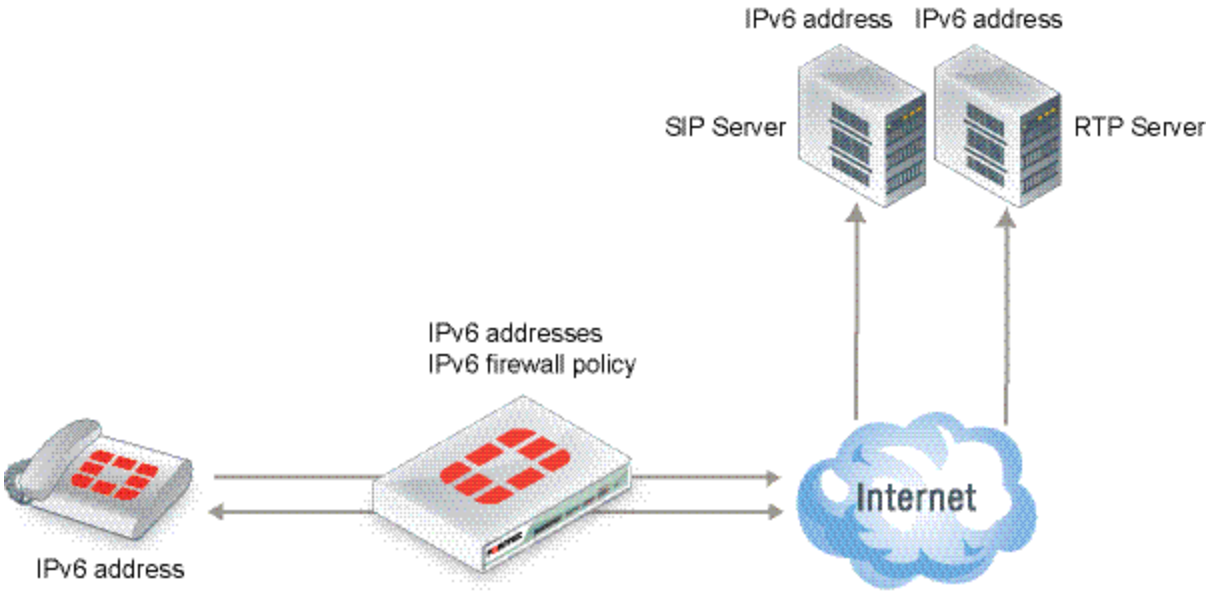
For configuration information, see IPv6 IPsec VPN on page 37.

SIP over IPv6

FortiOS supports Sessions Initiate Protocol (SIP) over IPv6. The SIP application-level gateway (ALG) can process SIP messages that use IPv6 addresses in the headers, bodies, and in the transport stack. The SIP ALG cannot modify the IPv6 addresses in the SIP headers so FortiGate units cannot perform SIP or RTP NAT over IPv6 and also cannot translate between IPv6 and IPv4 addresses.

In the scenario shown below, a SIP phone connects to the Internet through a FortiGate unit operating. The phone and the SIP and RTP servers all have IPv6 addresses.

The FortiGate unit has IPv6 security policies that accept SIP sessions. The SIP ALG understands IPv6 addresses and can forward IPv6 sessions to their destinations. Using SIP application control features the SIP ALG can also apply rate limiting and other settings to SIP sessions.



To enable SIP support for IPv6 add an IPv6 security policy that accepts SIP packets and includes a VoIP profile.

New Fortinet FortiGate IPv6 MIB fields

The following IPv6 MIB fields have been added to the Fortinet FortiGate MIB. These MIB entries can be used to display IPv6 session and policy statistics.

- IPv6 Session Counters:

```
fgSysSes6Count
fgSysSes6Rate1
fgSysSes6Rate10
fgSysSes6Rate30
fgSysSes6Rate60
```

- IPv6 Policy Statistics:

```
fgFwPol6StatsTable
fgFwPol6StatsEntry
FgFwPol6StatsEntry
fgFwPol6ID
fgFwPol6PktCount
fgFwPol6ByteCount
```

- IPv6 Session Statistics:

```
fgIp6SessStatsTable
fgIp6SessStatsEntry
FgIp6SessStatsEntry
fgIp6SessNumber
```

The fgSysSesCount and fgSysSesRateX MIBs report statistics for IPv4 plus IPv6 sessions combined. This behavior was not changed.

New OIDs

The following OIDs have been added:

```
FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgSystem.fgSystemInfo
.fgSysSes6Count   1.3.6.1.4.1.12356.101.4.1.15
.fgSysSesRate1    1.3.6.1.4.1.12356.101.4.1.16
.fgSysSesRate10   1.3.6.1.4.1.12356.101.4.1.17
.fgSysSesRate30   1.3.6.1.4.1.12356.101.4.1.18
.fgSysSesRate60   1.3.6.1.4.1.12356.101.4.1.19

FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgFirewall.fgFwPolicies.fgFwPolTables
.fgFwPol6StatsTable.fgFwPol6StatsEntry.fgFwPol6ID   1.3.6.1.4.1.12356.101.5.1.2.2.1.1
.fgFwPol6StatsTable.fgFwPol6StatsEntry.fgFwPol6PktCount   1.3.6.1.4.1.12356.101.5.1.2.2.1.
    2
.fgFwPol6StatsTable.fgFwPol6StatsEntry.fgFwPol6ByteCount   1.3.6.1.4.1.12356.101.5.1.2.2.1
    .3

FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgInetProto.fgInetProtoTables
.fgIp6SessStatsTable.fgIp6SessStatsEntry.fgIp6SessNumber 1.3.6.1.4.1.12356.101.11.2.3.1.1
```

EXAMPLE SNMP get/walk output

```
// Session6 stats excerpt from sysinfo:
snmpwalk -v2c -cpublic 192.168.1.111 1.3.6.1.4.1.12356.101.4
FORTINET-FORTIGATE-MIB::fgSysSes6Count.0 = Gauge32: 203
FORTINET-FORTIGATE-MIB::fgSysSes6Rate1.0 = Gauge32: 10 Sessions Per Second
FORTINET-FORTIGATE-MIB::fgSysSes6Rate10.0 = Gauge32: 2 Sessions Per Second
FORTINET-FORTIGATE-MIB::fgSysSes6Rate30.0 = Gauge32: 1 Sessions Per Second
FORTINET-FORTIGATE-MIB::fgSysSes6Rate60.0 = Gauge32: 0 Sessions Per Second
```

```
// FwPolicy6 table:
snmpwalk -v2c -cpublic 192.168.1.111 1.3.6.1.4.1.12356.101.5.1.2.2
FORTINET-FORTIGATE-MIB::fgFwPol6ID.1.3 = INTEGER: 3
FORTINET-FORTIGATE-MIB::fgFwPol6ID.1.4 = INTEGER: 4
FORTINET-FORTIGATE-MIB::fgFwPol6PktCount.1.3 = Counter64: 4329
FORTINET-FORTIGATE-MIB::fgFwPol6PktCount.1.4 = Counter64: 0
FORTINET-FORTIGATE-MIB::fgFwPol6ByteCount.1.3 = Counter64: 317776
FORTINET-FORTIGATE-MIB::fgFwPol6ByteCount.1.4 = Counter64: 0

// IP6SessNumber:
snmpwalk -v2c -cpublic 192.168.1.111 1.3.6.1.4.1.12356.101.11.2.3.1
FORTINET-FORTIGATE-MIB::fgIp6SessNumber.1 = Counter32: 89
```

IPv6 Per-IP traffic shaper

You can add any Per-IP traffic shaper to an IPv6 security policy using the following command:

```
config firewall policy6
   edit 0
      set per-ip-shaper "new-perip-shaper"
   end
```

DHCPv6

You can use DHCP with IPv6 using the CLI. To configure DHCP, ensure IPv6 is enabled by going to **System > Config > Features** and enabling **IPv6**.

Use the CLI command

```
config system dhcp6
```

For more information on the configuration options, see the FortiGate CLI Reference.

DHCPv6 relay

You can use the following command to configure a FortiGate interface to relay DHCPv6 queries and responses from one network to a network with a DHCPv6 server and back. The command enables DHCPv6 relay and includes adding the IPv6 address of the DHCP server that the FortiGate unit relays DHCPv6 requests to:

```
config system interface
   edit internal
      config ipv6
         set dhcp6-relay-service enable
         set dhcp6-relay-type regular
         set dhcp6-relay-ip 2001:db8:0:2::30
      end
```

IPv6 forwarding

Policies, IPS, Application Control, flow-based antivirus, web filtering, and DLP

FortiOS fully supports flow-based inspection of IPv6 traffic. This includes full support for IPS, application control, virus scanning, and web filtering.
To add flow-based inspection to IPv6 traffic go to **Policy & Objects > Policy > IPv6** and select **Create New** to add an IPv6 Security Policy. Configure the policy to accept the traffic to be scanned. Under **Security Profiles**, select the profiles to apply to the traffic.

Obtaining IPv6 addresses from an IPv6 DHCP server

From the CLI, you can configure any FortiGate interface to get an IPv6 address from an IPv6 DHCP server. For example, to configure the wan2 interface to get an IPv6 address from an IPv6 DHCP server enter the following command:

```
config system interface
   edit wan2
      config ipv6
         set ip6-mode dhcp
      end
```

IPv6 Configuration

This section contains configuration information for IPv6 on FortiOS. Attempts are made to include scenarios in each section to better assist with the configuration and to orient the information toward a particular task.

You will find information on the following:

By default IPv6 configurations do not appear in the web-based manager. You need to enable the feature first.

To enable IPv6:

1. Go to **System > Config > Features**.
2. Select **IPv6** and click **Apply**.

IPv6 address groups

To create IPv6 address groups from existing IPv6 addresses - web-based manager

Your company has 3 internal servers with IPv6 addresses that it would like to group together for the purposes of a number of policies.

1. Go to **Policy & Objects > Objects > Addresses** and select **Create New > Address Group.**
2. Select **IPv6 Group**, and fill out the fields with the following information:

Group Name	Web_Server_Cluster

Members	Web_Server-1
	Web_Server-2
	Web_Server-3

3. Select **OK.**

To create IPv6 address groups from existing IPv6 addresses - CLI

```
config firewall addrgrp6
   edit Web_Server_Cluster
      set member Web_Server-1 Web_Server-2 Web_Server-3
   end
```

To verify that the addresses were added correctly

1. Go to **Policy & Objects > Objects > Addresses.** Check that the addresses have been added to the address list and that they are correct.
2. From the CLI, enter the following commands:

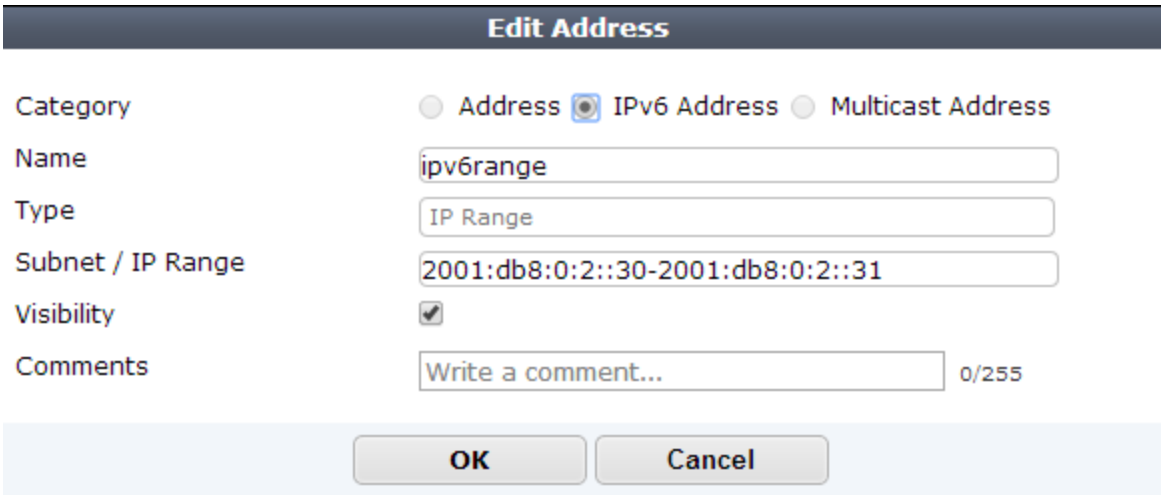
```
config firewall addgrp6
   edit <the name of the address that you wish to verify>
      Show full-configuration
```

IPv6 address ranges

You can configure IPv6 address ranges in both the GUI and the CLI.

To configure IPv6 address ranges - web-based manager:

1. Go to **Policy & Objects > Objects > Addresses.**
2. Set the **Type** to **IP Range** and enter the IPv6 addresses as shown:



To configure IPv6 address ranges - CLI:

```
config firewall address6
   edit ipv6range
      set type iprange
      set start-ip 2001:db8:0:2::30
      set end-ip 2001:db8:0:2::31
   end
```

IPv6 firewall addresses

Scenario: Mail Server

You need to create an IPv6 address for the Mail Server on Port1 of your internal network. This server is on the network off of port1.

- The IP address is 2001:db8:0:2::20/128
- There should be a tag for this address being for a server.

Configuring the Example using the GUI

1. Go to **Policy & Objects > Objects > Addresses** and select **Create New > Address.**
2. Select **IPv6 Address** and fill out the fields with the following information

Name	Mail_Server
Type	Subnet
IPv6 Address	2001:db8:0:2::20/128

3. Select **OK.**

Configuring the Example using the CLI

Enter the following CLI command:

```
config firewall address6
   edit Mail_Server
      set type ipprefix
      set subnet 2001:db8:0:2::20/128
   end
```

Scenario: First Floor Network

You need to create an IPv6 address for the subnet of the internal network off of Port1. These computers connect to port1. The network uses the IPv6 addresses: fdde:5a7d:f40b:2e9d:xxxx:xxxx:xxxx:xxxx

There should be a reference to this being the network for the 1st floor of the building.

1. Go to **Policy & Objects > Objects > Addresses**
2. Select **Create New > Address.**Select **IPv6 Address** and fill out the fields with the following information:

Name	Internal_Subnet_1
Type	Subnet / IP Range
IPv6 Address	2001:db8:0:2::/64
Comments	Network for 1st Floor

3. Select **OK.**
4. Enter the following CLI command:

```
config firewall address6
   edit Internal_Subnet_1
      set comment "Network for 1st Floor"
      set type ipprefix
      set subnet 2001:db8:0:2::/64
   end
```

Scenario: Accounting Team

You need to create an IPv6 address for the Accounting Team that's on the 1st Floor. These users are off of various ports of the FortiGate, but they have all been assigned addresses between 2001:db8:0:2::2000 and 2001:db8:0:2::a000

Configuring the Example using the GUI

1. Go to **Policy & Objects > Objects > Addresses** and select **Create New > Address.**
2. Select **IPv6 Address** and fill out the fields with the following information

Name	Accounting_Team
Type	IP Range
Subnet / IP Range	2001:db8:0:2::2000-2001:db8:0:2::a000

3. Select **OK.**

Configuring the Example using the CLI

Enter the following CLI command:

```
config firewall address6
   edit Accounting_Team
      set type iprange
      set visibility enable
      set start-ip 2001:db8:0:2::2000
      set end-ip 2001:db8:0:2::a000
      end
```

To verify that the addresses were added correctly:

1. Go to **Policy & Objects > Objects > Addresses.** Check that the addresses have been added to the address list and that they are correct.
2. Enter the following CLI command:

```
config firewall address6
   edit <the name of the address that you wish to verify>
      Show full-configuration
```

ICMPv6

The IT Manager is doing some diagnostics and would like to temporarily block the successful replies of ICMP Node information Responses between 2 IPv6 networks.

The ICMP type for ICMP Node informations responses is 140. The codes for a successful response is 0.

To configure ICMPv6 - web-based manager:

1. Go to **Policy & Objects > Objects > Services** and select **Create New > Service**.
2. Fill out the fields with the following information

Name	diagnostic-test1
Service Type	Firewall
Show in Service List	Enabled
Category	Uncategorized
Protocol Type	ICMP6
Type	140

3. Select **OK.**
4. Enter the following CLI command:

```
config firewall service custom
   edit diagnostic-test1
      set protocol ICMP6
      set icmptype 140
      set icmpcode 0
      set visibility enable
   end
```

To verify that the category was added correctly:

1. Go to **Policy & Objects > Objects > Services.**Check that the services have been added to the services list and that they are correct.
2. Enter the following CLI command:

```
config firewall service custom
   edit <the name of the service that you wish to verify>
      show full-configuration
```

IPv6 IPsec VPN

This chapter describes how to configure your FortiGate unit's IPv6 IPsec VPN functionality.

By default IPv6 configurations do not appear in the web-based manager. You need to enable the feature first.

To enable IPv6:

1. Go to **System > Config > Features**.
2. Select **IPv6** and click **Apply**.

The topics in this section include:

Overview of IPv6 IPsec support

FortiOS supports route-based IPv6 IPsec, but not policy-based. This section describes how IPv6 IPsec support differs from IPv4 IPsec support.

Where both the gateways and the protected networks use IPv6 addresses, sometimes called IPv6 over IPv6, you can create either an auto-keyed or manually-keyed VPN. You can also combine IPv6 and IPv4 addressing in an auto-keyed VPN in the following ways:

IPv4 over IPv6	The VPN gateways have IPv6 addresses. The protected networks have IPv4 addresses. The phase 2 configurations at either end use IPv4 selectors.
IPv6 over IPv4	The VPN gateways have IPv4 addresses. The protected networks use IPv6 addresses. The phase 2 configurations at either end use IPv6 selectors.

Compared with IPv4 IPsec VPN functionality, there are some limitations:

- Except for IPv6 over IPv4, remote gateways with Dynamic DNS are not supported.
- Selectors cannot be firewall address names. Only IP address, address range and subnet are supported.
- Redundant IPv6 tunnels are not supported.

Certificates

On a VPN with IPv6 phase 1 configuration, you can authenticate using VPN certificates in which the common name (cn) is an IPv6 address. The `cn-type` keyword of the `user peer` command has an option, `ipv6`, to support this.

Configuring IPv6 IPsec VPNs

Configuration of an IPv6 IPsec VPN follows the same sequence as for an IPv4 route-based VPN: phase 1 settings, phase 2 settings, security policies, and routing.

Phase 1 configuration

In the web-based manager, you define the Phase 1 as IPv6 in the Advanced settings. Enable the IPv6 Version check box. You can then enter an IPv6 address for the remote gateway.

In the CLI, you define an IPsec phase 1 configuration as IPv6 by setting `ip-version` to `6`. Its default value is `4`. Then, the `local-gw` and `remote-gw` keywords are hidden and the corresponding `local-gw6` and `remote-gw6` keywords are available. The values for `local-gw6` and `remote-gw6` must be IPv6 addresses. For example:

```
config vpn ipsec phase1-interface
   edit tunnel6
      set ip-version 6
      set remote-gw6 0:123:4567::1234
      set interface port3
      set proposal 3des-md5
   end
```

Phase 2 configuration

To create an IPv6 IPsec phase 2 configuration in the web-based manager, you need to define IPv6 selectors in the Advanced settings. Change the default “0.0.0.0/0” address for Source address and Destination address to the IPv6 value “::/0”. If needed, enter specific IPv6 addresses, address ranges or subnet addresses in these fields.

In the CLI, set `src-addr-type` and `dst-addr-type` to `ip6`, `range6` or `subnet6` to specify IPv6 selectors. By default, zero selectors are entered, “::/0” for the `subnet6` address type, for example. The simplest IPv6 phase 2 configuration looks like this:

```
config vpn ipsec phase2-interface
   edit tunnel6_p2
      set phase1name tunnel6
      set proposal 3des-md5
      set src-addr-type subnet6
      set dst-addr-type subnet6
   end
```

Security policies

To complete the VPN configuration, you need a security policy in each direction to permit traffic between the protected network’s port and the IPsec interface. You need IPv6 policies unless the VPN is IPv4 over IPv6.

Routing

Appropriate routing is needed for both the IPsec packets and the encapsulated traffic within them. You need a route, which could be the default route, to the remote VPN gateway via the appropriate interface. You also need a route to the remote protected network via the IPsec interface.

To create a static route - web-based manager:

1. Go to **Router > Static > Static Routes**.

On some desktop FortiGate models, go to **System > Network > Routing**.

2. Select the drop-down arrow on the **Create New** button and select **IPv6 Route**.
3. Enter the information and select **OK**.

To create a static route - CLI:

1. In the CLI, use the `router static6` command. For example, where the remote network is `fec0:0000:0000:0004::/64` and the IPsec interface is `toB`:

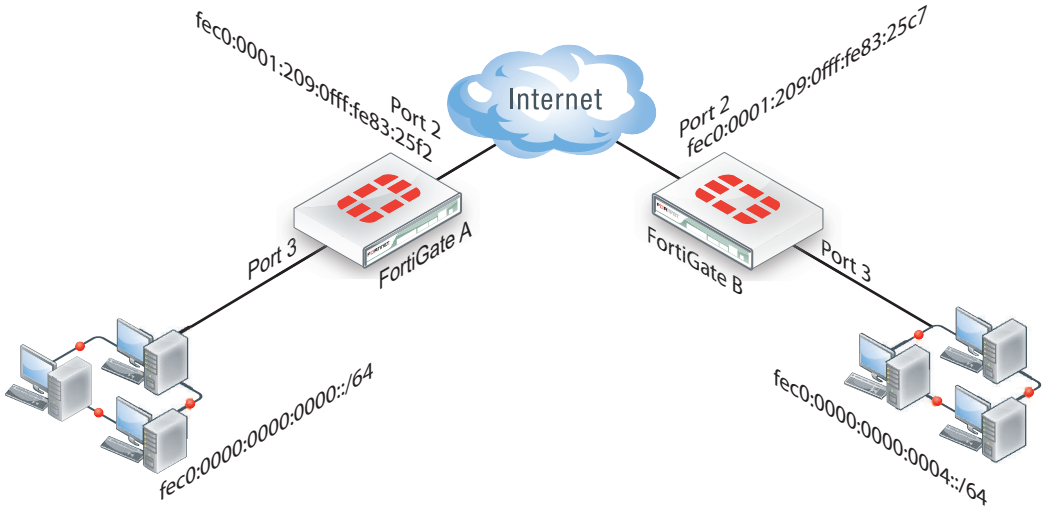
```
config router static6
   edit 1
      set device port2
      set dst 0::/0
   next
   edit 2
      set device toB
      set dst fec0:0000:0000:0004::/64
   next
end
```

If the VPN is IPV4 over IPv6, the route to the remote protected network is an IPv4 route. If the VPN is IPv6 over IPv4, the route to the remote VPN gateway is an IPv4 route.

Site-to-site IPv6 over IPv6 VPN example

In this example, computers on IPv6-addressed private networks communicate securely over public IPv6 infrastructure.

Example IPv6-over-IPv6 VPN topology



Configure FortiGate A interfaces

Port 2 connects to the public network and port 3 connects to the local network.

```
config system interface
   edit port2
      config ipv6
         set ip6-address fec0::0001:209:0fff:fe83:25f2/64
```

```
      end
   next
   edit port3
      config ipv6
         set ip6-address fec0::0000:209:0fff:fe83:25f3/64
      end
   next
end
```

Configure FortiGate A IPsec settings

The phase 1 configuration creates a virtual IPsec interface on port 2 and sets the remote gateway to the public IP address FortiGate B. This configuration is the same as for an IPv4 route-based VPN, except that `ip-version` is set to `6` and the `remote-gw6` keyword is used to specify an IPv6 remote gateway address.

```
config vpn ipsec phase1-interface
   edit toB
      set ip-version 6
      set interface port2
      set remote-gw6 fec0:0000:0000:0003:209:0fff:fe83:25c7
      set dpd enable
      set psksecret maryhadalittlelamb
      set proposal 3des-md5 3des-sha1
   end
```

By default, phase 2 selectors are set to accept all subnet addresses for source and destination. The default setting for `src-addr-type` and `dst-addr-type` is `subnet`. The IPv6 equivalent is `subnet`. The default subnet addresses are 0.0.0.0/0 for IPv4, `::/0` for IPv6.

```
config vpn ipsec phase2-interface
   edit toB2
      set phase1name toB
      set proposal 3des-md5 3des-sha1
      set pfs enable
      set replay enable
      set src-addr-type subnet6
      set dst-addr-type subnet6
   end
```

Configure FortiGate A security policies

Security policies are required to allow traffic between port3 and the IPsec interface toB in each direction. The address `all6` must be defined using the `firewall address6` command as `::/0`.

```
config firewall policy6
   edit 1
      set srcintf port3
      set dstintf toB
      set srcaddr all6
      set dstaddr all6
      set action accept
      set service ANY
      set schedule always
   next
   edit 2
      set srcintf toB
      set dstintf port3
      set srcaddr all6
      set dstaddr all6
```

```
      set action accept
      set service ANY
      set schedule always
   end
```

Configure FortiGate A routing

This simple example requires just two static routes. Traffic to the protected network behind FortiGate B is routed via the virtual IPsec interface toB. A default route sends all IPv6 traffic out on port2.

```
config router static6
   edit 1
      set device port2
      set dst 0::/0
   next
   edit 2
      set device toB
      set dst fec0:0000:0000:0004::/64
   end
```

Configure FortiGate B

The configuration of FortiGate B is very similar to that of FortiGate A. A virtual IPsec interface toA is configured on port2 and its remote gateway is the public IP address of FortiGate A. Security policies enable traffic to pass between the private network and the IPsec interface. Routing ensures traffic for the private network behind FortiGate A goes through the VPN and that all IPv6 packets are routed to the public network.

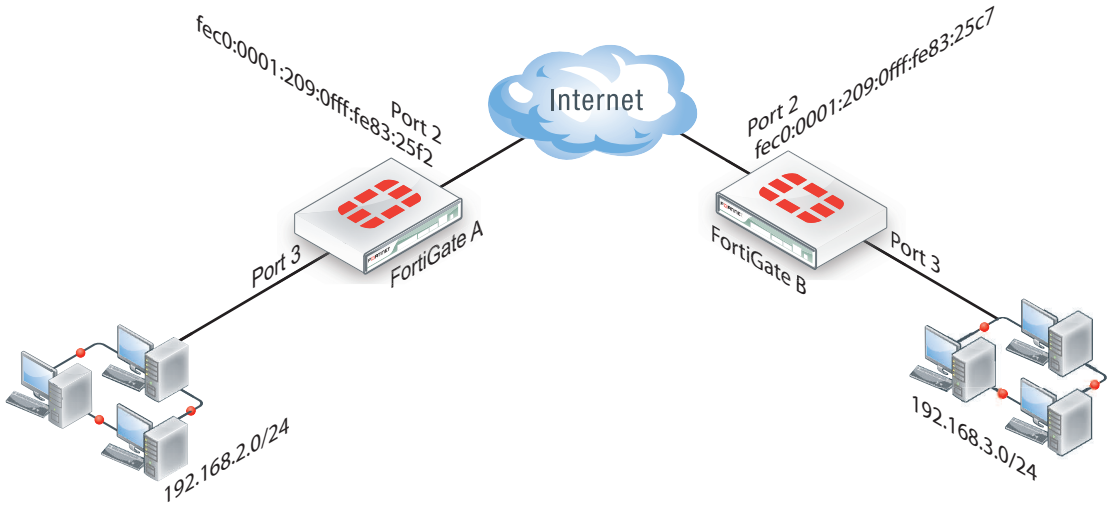
```
config system interface
   edit port2
      config ipv6
         set ip6-address fec0::0003:209:0fff:fe83:25c7/64
      end
   next
   edit port3
      config ipv6
         set ip6-address fec0::0004:209:0fff:fe83:2569/64
      end
   end
config vpn ipsec phase1-interface
   edit toA
      set ip-version 6
      set interface port2
      set remote-gw6 fec0:0000:0000:0001:209:0fff:fe83:25f2
      set dpd enable
      set psksecret maryhadalittlelamb
      set proposal 3des-md5 3des-sha1
   end
config vpn ipsec phase2-interface
   edit toA2
      set phase1name toA
      set proposal 3des-md5 3des-sha1
      set pfs enable
      set replay enable
      set src-addr-type subnet6
      set dst-addr-type subnet6
   end
config firewall policy6
   edit 1
```

```
        set srcintf port3
        set dstintf toA
        set srcaddr all6
        set dstaddr all6
        set action accept
        set service ANY
        set schedule always
    next
    edit 2
        set srcintf toA
        set dstintf port3
        set srcaddr all6
        set dstaddr all6
        set action accept
        set service ANY
        set schedule always
    end
config router static6
    edit 1
        set device port2
        set dst 0::/0
    next
    edit 2
        set device toA
        set dst fec0:0000:0000:0000::/64
end
```

Site-to-site IPv4 over IPv6 VPN example

In this example, two private networks with IPv4 addressing communicate securely over IPv6 infrastructure.

Example IPv4-over-IPv6 VPN topology



Configure FortiGate A interfaces

Port 2 connects to the IPv6 public network and port 3 connects to the IPv4 LAN.

```
config system interface
```

```
   edit port2
      config ipv6
         set ip6-address fec0::0001:209:0fff:fe83:25f2/64
      end
   next
   edit port3
      set 192.168.2.1/24
   end
```

Configure FortiGate A IPsec settings

The phase 1 configuration is the same as in the IPv6 over IPv6 example.

```
config vpn ipsec phase1-interface
   edit toB
      set ip-version 6
      set interface port2
      set remote-gw6 fec0:0000:0000:0003:209:0fff:fe83:25c7
      set dpd enable
      set psksecret maryhadalittlelamb
      set proposal 3des-md5 3des-sha1
   end
```

The phase 2 configuration is the same as you would use for an IPv4 VPN. By default, phase 2 selectors are set to accept all subnet addresses for source and destination.

```
config vpn ipsec phase2-interface
   edit toB2
      set phase1name toB
      set proposal 3des-md5 3des-sha1
      set pfs enable
      set replay enable
   end
```

Configure FortiGate A security policies

Security policies are required to allow traffic between port3 and the IPsec interface toB in each direction. These are IPv4 security policies.

```
config firewall policy
   edit 1
      set srcintf port3
      set dstintf toB
      set srcaddr all
      set dstaddr all
      set action accept
      set service ANY
      set schedule always
   next
   edit 2
      set srcintf toB
      set dstintf port3
      set srcaddr all
      set dstaddr all
      set action accept
      set service ANY
      set schedule always
   end
```

Configure FortiGate A routing

This simple example requires just two static routes. Traffic to the protected network behind FortiGate B is routed via the virtual IPsec interface toB using an IPv4 static route. A default route sends all IPv6 traffic, including the IPv6 IPsec packets, out on port2.

```
config router static6
   edit 1
      set device port2
      set dst 0::/0
   next
   edit 2
      set device toB
      set dst 192.168.3.0/24
   end
```

Configure FortiGate B

The configuration of FortiGate B is very similar to that of FortiGate A. A virtual IPsec interface toA is configured on port2 and its remote gateway is the public IP address of FortiGate A. The IPsec phase 2 configuration has IPv4 selectors.

IPv4 security policies enable traffic to pass between the private network and the IPsec interface. An IPv4 static route ensures traffic for the private network behind FortiGate A goes through the VPN and an IPv6 static route ensures that all IPv6 packets are routed to the public network.

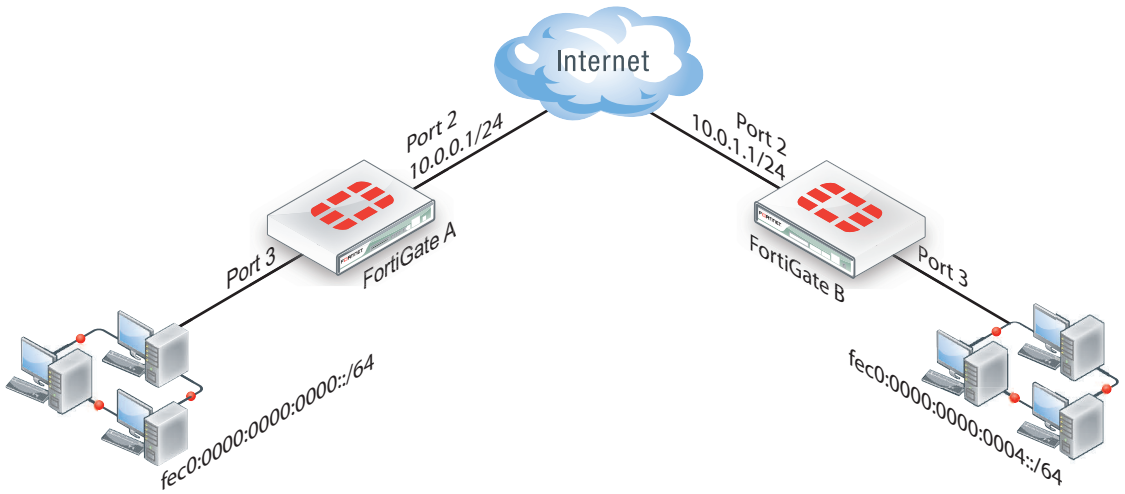
```
config system interface
   edit port2
      config ipv6
         set ip6-address fec0::0003:fe83:25c7/64
      end
   next
   edit port3
      set 192.168.3.1/24
   end
config vpn ipsec phase1-interface
   edit toA
      set ip-version 6
      set interface port2
      set remote-gw6 fec0:0000:0000:0001:209:0fff:fe83:25f2
      set dpd enable
      set psksecret maryhadalittlelamb
      set proposal 3des-md5 3des-sha1
   end
config vpn ipsec phase2-interface
   edit toA2
      set phase1name toA
      set proposal 3des-md5 3des-sha1
      set pfs enable
      set replay enable
   end
config firewall policy
   edit 1
      set srcintf port3
      set dstintf toA
      set srcaddr all
      set dstaddr all
```

```
        set action accept
        set service ANY
        set schedule always
    next
    edit 2
        set srcintf toA
        set dstintf port3
        set srcaddr all
        set dstaddr all
        set action accept
        set service ANY
        set schedule always
    end
config router static6
    edit 1
        set device port2
        set dst 0::/0
    next
    edit 2
        set device toA
        set dst 192.168.2.0/24
    end
```

Site-to-site IPv6 over IPv4 VPN example

In this example, IPv6-addressed private networks communicate securely over IPv4 public infrastructure.

Example IPv6-over-IPv4 VPN topology



Configure FortiGate A interfaces

Port 2 connects to the IPv4 public network and port 3 connects to the IPv6 LAN.

```
config system interface
    edit port2
        set 10.0.0.1/24
        next
        edit port3
        config ipv6
```

```
      set ip6-address fec0::0001:209:0fff:fe83:25f3/64
   end
```

Configure FortiGate A IPsec settings

The phase 1 configuration uses IPv4 addressing.

```
config vpn ipsec phase1-interface
   edit toB
      set interface port2
      set remote-gw 10.0.1.1
      set dpd enable
      set psksecret maryhadalittlelamb
      set proposal 3des-md5 3des-sha1
   end
```

The phase 2 configuration uses IPv6 selectors. By default, phase 2 selectors are set to accept all subnet addresses for source and destination. The default setting for `src-addr-type` and `dst-addr-type` is `subnet`. The IPv6 equivalent is `subnet6`. The default subnet addresses are 0.0.0.0/0 for IPv4, `::/0` for IPv6.

```
config vpn ipsec phase2-interface
   edit toB2
      set phase1name toB
      set proposal 3des-md5 3des-sha1
      set pfs enable
      set replay enable
      set src-addr-type subnet6
      set dst-addr-type subnet6
   end
```

Configure FortiGate A security policies

IPv6 security policies are required to allow traffic between port3 and the IPsec interface toB in each direction. Define the address `all6` using the `firewall address6` command as `::/0`.

```
config firewall policy6
   edit 1
      set srcintf port3
      set dstintf toB
      set srcaddr all6
      set dstaddr all6
      set action accept
      set service ANY
      set schedule always
   next
   edit 2
      set srcintf toB
      set dstintf port3
      set srcaddr all6
      set dstaddr all6
      set action accept
      set service ANY
      set schedule always
   end
```

Configure FortiGate A routing

This simple example requires just two static routes. Traffic to the protected network behind FortiGate B is routed via the virtual IPsec interface toB using an IPv6 static route. A default route sends all IPv4 traffic, including the IPv4 IPsec packets, out on port2.

```
config router static6
   edit 1
      set device toB
      set dst fec0:0000:0000:0004::/64
   end
config router static
   edit 1
      set device port2
      set dst 0.0.0.0/0
      set gateway 10.0.0.254
   end
```

Configure FortiGate B

The configuration of FortiGate B is very similar to that of FortiGate A. A virtual IPsec interface toA is configured on port2 and its remote gateway is the IPv4 public IP address of FortiGate A. The IPsec phase 2 configuration has IPv6 selectors.

IPv6 security policies enable traffic to pass between the private network and the IPsec interface. An IPv6 static route ensures traffic for the private network behind FortiGate A goes through the VPN and an IPv4 static route ensures that all IPv4 packets are routed to the public network.

```
config system interface
   edit port2
      set 10.0.1.1/24
   next
   edit port3
      config ipv6
         set ip6-address fec0::0004:209:0fff:fe83:2569/64
      end
config vpn ipsec phase1-interface
   edit toA
      set interface port2
      set remote-gw 10.0.0.1
      set dpd enable
      set psksecret maryhadalittlelamb
      set proposal 3des-md5 3des-sha1
   end
config vpn ipsec phase2-interface
   edit toA2
      set phase1name toA
      set proposal 3des-md5 3des-sha1
      set pfs enable
      set replay enable
      set src-addr-type subnet6
      set dst-addr-type subnet6
   end
config firewall policy6
   edit 1
      set srcintf port3
      set dstintf toA
      set srcaddr all6
      set dstaddr all6
      set action accept
      set service ANY
      set schedule always
   next
   edit 2
      set srcintf toA
      set dstintf port3
      set srcaddr all6
      set dstaddr all6
      set action accept
      set service ANY
      set schedule always
   end
config router static6
   edit 1
      set device toA
```

```
      set dst fec0:0000:0000:0000::/64
   end
config router static
   edit 1
      set device port2
      set gateway 10.0.1.254
   end
```

TCP MSS values

TCP MSS values, which control the maximum amount of data that can be sent in a single packet, can be set for IPv6 policies (for both the sender and the receiver). You can configure TCP MSS values in IPv6 using the following CLI commands:

```
config firewall policy6
   edit <index_int>
      set tcp-mss-sender <value>
      set tcp-mss-receiver <value>
   end
```

BGP and IPv6

FortiGate units support IPv6 over BGP using the same `config router bgp` command as IPv4, but different subcommands.

The main CLI keywords have IPv6 equivalents that are identified by the “6” on the end of the keyword, such as with `config network6` or `set allowas-in6`.

IPv6 BGP commands include:

```
config router bgp
   set activate6 {enable | disable}
   set allowas-in6 <max_num_AS_integer>
   set allowas-in-enable6 {enable | disable}
   set as-override6 {enable | disable}
   set attribute-unchanged6 [as-path] [med] [next-hop]
   set capability-default-originate6 {enable | disable}
   set capability-graceful-restart6 {enable | disable}
   set default-originate-route-map6 <routemap_str>
   set distribute-list-in6 <access-list-name_str>
   set distribute-list-out6 <access-list-name_str>
   set filter-list-in6 <aspath-list-name_str>
   set filter-list-out6 <aspath-list-name_str>
   set maximum-prefix6 <prefix_integer>
   set maximum-prefix-threshold6 <percentage_integer>
   set maximum-prefix-warning-only6 {enable | disable}
   set next-hop-self6 {enable | disable}
   set prefix-list-in6 <prefix-list-name_str>
   set prefix-list-out6 <prefix-list-name_str>
   set remove-private-as6 {enable | disable}
   set route-map-in6 <routemap-name_str>
   set route-map-out6 <routemap-name_str>
   set route-reflector-client6 {enable | disable}
```

```
    set route-server-client6 {enable | disable}
    set send-community6 {both | disable | extended | standard}
    set soft-reconfiguration6 {enable | disable}
    set unsuppress-map6 <route-map-name_str>
    config network6
    config redistribute6
end
```

RIPng — RIP and IPv6

RIP next generation, or RIPng, is the version of RIP that supports IPv6.

This is an example of a typical small network configuration using RIPng routing.

Your internal R&D network is working on a project for a large international telecom company that uses IPv6. For this reason, you have to run IPv6 on your internal network and you have decided to use only IPv6 addresses.

Your network has two FortiGate units running the RIPng dynamic routing protocol. Both FortiGate units are connected to the ISP router and the internal network. This configuration provides some redundancy for the R&D internal network enabling it to reach the internet at all times.

This section includes the following topics:

Network layout and assumptions

Basic network layout

Your internal R&D network is working on a project for a large international telecom company that uses IPv6. For this reason, you have to run IPv6 on your internal network and you have decided to use only IPv6 addresses.

Your network has two FortiGate units running the RIPng dynamic routing protocol. Both FortiGate units are connected to the ISP router and the internal network. This configuration provides some redundancy for the R&D internal network enabling it to reach the internet at all times.

All internal computers use RIP routing, so no static routing is required. And all internal computers use IPv6 addresses.

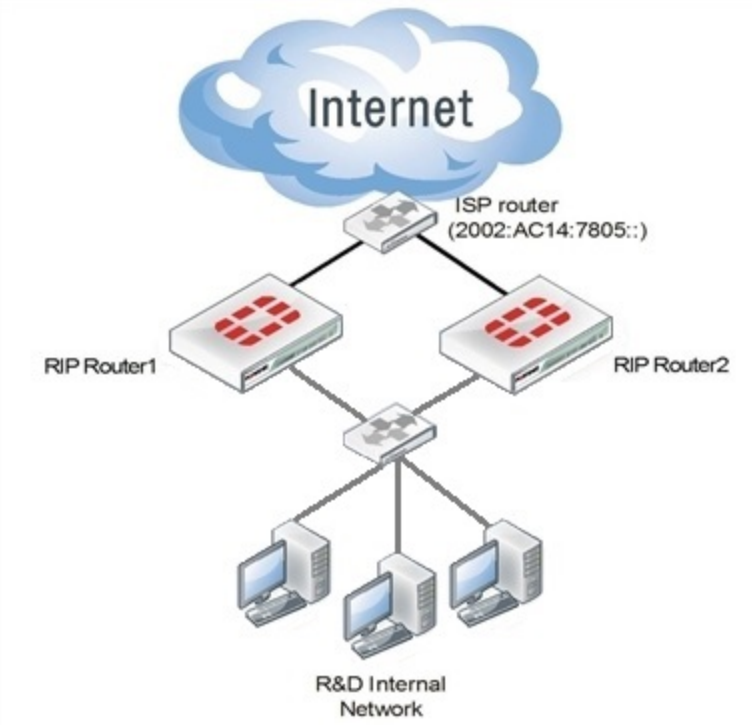
Where possible in this example, the default values will be used or the most general settings. This is intended to provide an easier configuration that will require less troubleshooting.

In this example the routers, networks, interfaces used, and IP addresses are as follows:

Rip example network topology

Network	Router	Interface & Alias	IPv6 address
R&D	Router1	port1 (internal)	2002:A0B:6565:0:0:0:0:0
		port2 (ISP)	2002:AC14:7865:0:0:0:0:0
	Router2	port1 (internal)	2002:A0B:6566:0:0:0:0:0
		port2 (ISP)	2002:AC14:7866:0:0:0:0:0

Network topology for the IPv6 RIPng example



Assumptions

The following assumptions have been made concerning this example:

- All FortiGate units have 5.0+ firmware, and are running factory default settings.
- All CLI and web-based manager navigation assumes the unit is running in NAT/Route operating mode, with VDOMs disabled.
- All FortiGate units have interfaces labelled port1 and port2 as required.
- All firewalls have been configured for each FortiGate unit to allow the required traffic to flow across interfaces.
- All network devices are support IPv6 and are running RIPng.

Configuring the FortiGate units system information

Each FortiGate unit needs IPv6 enabled, a new hostname, and interfaces configured.

To configure system information on Router1 - web-based manager:

1. Go to **System > Dashboard > Status**.
2. For **Host name**, select **Change**.
3. Enter “Router1”.
4. Go to **System > Config > Features**.
5. Enable **IPv6** and click **Apply**.
6. Go to **System > Network > Interfaces**.
7. Edit port1 (internal) interface.
8. Set the following information, and select **OK**.

Alias	internal
IP/Netmask	2002:A0B:6565::/0
Administrative Access	HTTPS SSH PING
Description	Internal RnD network
Administrative Status	Up

9. Edit port2 (ISP) interface.
10. Set the following information, and select **OK**.

Alias	ISP
IP/Netmask	2002:AC14:7865::/0
Administrative Access	HTTPS SSH PING
Description	ISP and internet
Administrative Status	Up

To configure system information on Router1 - CLI:

```
config system global
   set hostname Router1
   set gui-ipv6 enable
end
config system interface
   edit port1
      set alias internal
      set allowaccess https ping ssh
      set description "Internal RnD network"
      config ipv6
         set ip6-address 2002:a0b:6565::/0
      end
   next
   edit port2
      set alias ISP
      set allowaccess https ping ssh
      set description "ISP and internet"
```

```
      config ipv6
         set ip6-address 2002:AC14:7865::
      end
   end
```

To configure system information on Router2 - web-based manager:

1. Go to **System > Dashboard > Status**.
2. For **Host name**, select **Change**.
3. Enter “Router2”.
4. Go to **System > Config > Features**.
5. Enable **IPv6** and click **Apply**.
6. Go to **System > Network > Interfaces**.
7. Edit port1 (internal) interface.
8. Set the following information, and select **OK**.

Alias	internal
IP/Netmask	2002:A0B:6566::/0
Administrative Access	HTTPS SSH PING
Description	Internal RnD network
Administrative Status	Up

9. Edit port2 (ISP) interface.
10. Set the following information, and select **OK**.

Alias	ISP
IP/Netmask	2002:AC14:7866::/0
Administrative Access	HTTPS SSH PING
Description	ISP and internet
Administrative Status	Up

To configure system information on Router2 - CLI:

```
config system global
   set hostname Router2
   set gui-ipv6 enable
end
config system interface
   edit port1
      set alias internal
      set allowaccess https ping ssh
      set description "Internal RnD network"
      config ipv6
         set ip6-address 2002:a0b:6566::/0
      end
```

```
next
edit port2
   set alias ISP
   set allowaccess https ping ssh
   set description "ISP and internet"
   config ipv6
      set ip6-address 2002:AC14:7866::
   end
end
```

Configuring RIPng on FortiGate units

Now that the interfaces are configured, you can configure RIPng on the FortiGate units.

There are only two networks and two interfaces to include — the internal network, and the ISP network. There is no redistribution, and no authentication. In RIPng there is no specific command to include a subnet in the RIP broadcasts. There is also no information required for the interfaces beyond including their name.

As this is a CLI only configuration, configure the ISP router and the other FortiGate unit as neighbors. This was not part of the previous example as this feature is not offered in the web-based manager. Declaring neighbors in the configuration like this will reduce the discovery traffic when the routers start up.

Since RIPng is not supported in the web-based manager, this section will only be entered in the CLI.

To configure RIPng on Router1 - CLI:

```
config router ripng
   config interface
      edit port1
      next
      edit port2
      end
   config neighbor
      edit 1
         set interface port1
         set ipv6 2002:a0b:6566::/0
      next
      edit 2
         set interface port2
         set ipv6 2002:AC14:7805::/0
      end
```

To configure RIPng on Router2 - CLI:

```
config router ripng
   config interface
      edit port1
      next
      edit port2
      end
   config neighbor
      edit 1
         set interface port1
         set ipv6 2002:a0b:6565::/0
      next
      edit 2
         set interface port2
```

```
    set ipv6 2002:AC14:7805::/0
end
```

Configuring other network devices

The other devices on the internal network all support IPv6, and are running RIPng where applicable. They only need to know the internal interface network addresses of the FortiGate units.

The ISP routers need to know the FortiGate unit information such as IPv6 addresses.

Testing the configuration

In addition to normal testing of your network configuration, you must also test the IPv6 part of this example.

For troubleshooting problems with your network, see the FortiOS Handbook Troubleshooting chapter.

Testing the IPv6 RIPng information

There are some commands to use when checking that your RIPng information is correct on your network. These are useful to check on your RIPng FortiGate units on your network. Comparing the output between devices will help you understand your network better, and also track down any problems.

`diagnose ipv6 address list`

View the local scope IPv6 addresses used as next-hops by RIPng on the FortiGate unit.

`diagnose ipv6 route list`

View ipv6 addresses that are installed in the routing table.

`get router info6 routing-table`

View the routing table. This information is almost the same as the previous command (diagnose ipv6 route list) however it is presented in an easier to read format.

`get router info6 rip interface external`

View brief output on the RIP information for the interface listed. The information includes if the interface is up or down, what routing protocol is being used, and whether passive interface or split horizon are enabled.

`get router info6 neighbor-cache list`

View the IPv6/MAC address mapping. This also displays the interface index and name associated with the address.

Debugging IPv6 on RIPng

The debug commands are very useful to see what is happening on the network at the packet level. There are a few changes to debugging the packet flow when debugging IPv6.

The following CLI commands specify both IPv6 and RIP, so only RIPng packets will be reported. The output from these commands will show you the RIPng traffic on your FortiGate unit including RECV, SEND, and UPDATE actions.

The addresses are in IPv6 format.

```
diagnose debug enable
diagnose ipv6 router rip level info
diagnose ipv6 router rip all enable
```

These three commands will:

Turn on debugging in general

Set the debug level to information, a verbose reporting level

Turn on all RIP router settings

Part of the information displayed from the debugging is the metric (hop count). If the metric is 16, then that destination is unreachable since the maximum hop count is 15.

In general, you should see an update announcement, followed by the routing table being sent out, and a received reply in response.

IPv6 RSSO support

RADIUS Single Sign-On (RSSO) is supported in IPv6, but can only be configured in the CLI:

```
config firewall policy6
   edit <id>
      set rsso enable
      set fall-through-unathenticated enable
   end
```

IPv6 IPS

IPv6 IPS signature scan can be enabled by interface policy. The user can create an normal IPS sensor and assign it to the IPv6 interface policy.

```
config firewall interface-policy6
   edit 1
      set interface "port1"
      set srcaddr6 "all"
      set dstaddr6 "all"
      set service6 "ANY"
      set ips-sensor-status enable
      set ips-sensor "all_default"
   next
end
```

Blocking IPv6 packets by extension headers

FortiOS can now block IPv6 packets based on the extension headers, using the CLI syntax:

```
config firewall ipv6-eh-filter.
```

The following commands are now available:

- `set hop-opt {disable | enable}`: Block packets with Hop-by-Hop Options header.
- `set dest-opt {disable | enable}`: Block packets with Destination Options header.
- `set hdopt-type <integer>`: Block specific Hop-by-Hop and/or Destination Option types (maximum 7 types, each between 0 and 255).

- `set routing {disable | enable}`: Block packets with Routing header.
- `set routing-type <integar>`: Block specific Routing header types (maximum 7 types, each between 0 and 255).
- `set fragment {disable | enable}`: Block packets with Fragment header.
- `set auth {disable | enable}`: Block packets with Authentication header.
- `set no-next {disable | enable}`: Block packets with No Next header.

IPv6 Denial of Service policies

Denial of Service (DoS) policies can now be configured by going to **Policy & Objects > Policy > IPv6 DoS**. For more information, refer to the “Interface Policies” section of the FortiOS Handbook Firewall chapter.

Configure hosts in an SNMP v1/2c community to send queries or receive traps

When you add a host to an SNMP v1/2c community you can now decide whether the FortiGate unit will accept queries from the host or whether the FortiGate unit will send traps to the host. You can also configure the host for both traps and queries. You can add up to 16 IPv4 hosts and up to 16 IPv6 hosts.

Use the following command to add two hosts to an SNMP community:

```
config system snmp community
   config hosts
      edit 1
         set interface port1
         set ip 172.20.120.1
         set host-type query
      end
   config hosts6
      edit 1
         set interface port6
         set ip 2001:db8:0:2::30
         set host-type trap
      end
```

IPv6 PIM sparse mode multicast routing

FortiOS supports PIM sparse mode multicast routing for IPv6 multicast (multicast6) traffic and is compliant with RFC 4601. You can use the following command to configure IPv6 PIM sparse multicast routing.

```
config router multicast6
   set multicast-routing {enable | disable}
      config interface
         edit <interface-name>
            set hello-interval <1-65535 seconds>
            set hello-holdtime <1-65535 seconds>
      end
      config pim-sm-global
      config rp-address
         edit <index>
```

```
            set ipv6-address <ipv6-address>
        end
```

The following diagnose commands for IPv6 PIM sparse mode are also available:

```
diagnose ipv6 multicast status
diagnose ipv6 multicast vif
diagnose ipv6 multicast mroute
```

FORTINET®

High Performance Network Security